Birla Institute of Technology & Science, Pilani
2^{nd} Semester 2016-17 - CS F211 – Data Structures and Algorithms

Lab 11 (Evaluation 3): 20 April 2017

Time: 170 minutes **Marks: 20 + 10 = 30**

Instructions:

- *This test consists of two problems (Problem 1 and Problem 2) specified in two different files.*
- All input expressions should be read from stdin (scanf) and output should be printed on stdout (printf).
- For first 150 minutes, only a subset of test cases will be visible to students after submitting the code on the portal. Only in last 20 minutes, all test cases will be made visible.
- At the end of 170 minute period, the online system will stop evaluating the submissions but it will accept it for additional 5 minutes. At the end of 175 minute period, it will stop accepting the submissions.
- Only the last submission by the student for each problem will be considered for evaluation, irrespective of earlier correct submission.
- Assuming that a problem contains M marks, in case of (Run-error/Compiler-error/Timelimit-error, or any error), evaluation will be done for M/2 marks only.
- Total marks of each problem contains some marks for modularity and proper structuring of code.
- All submitted source code will be later checked manually by the instructor and final marks will be awarded. Any case of plagiarism and/or hard coding of test cases will fetch 0 marks for the problem/evaluation component.
- Make sure to return 0 from the main() function in case of normal termination.

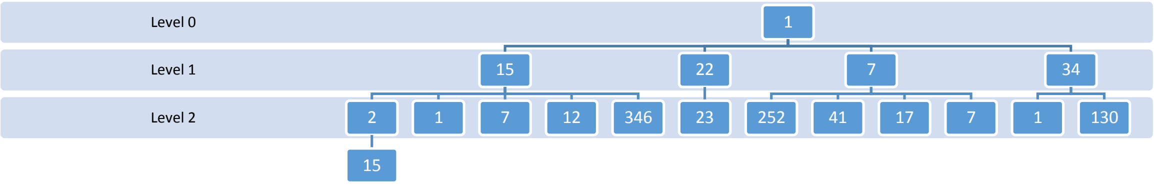
Problem 1 of 2

Expected Time: 110 minutes **Marks: 10**

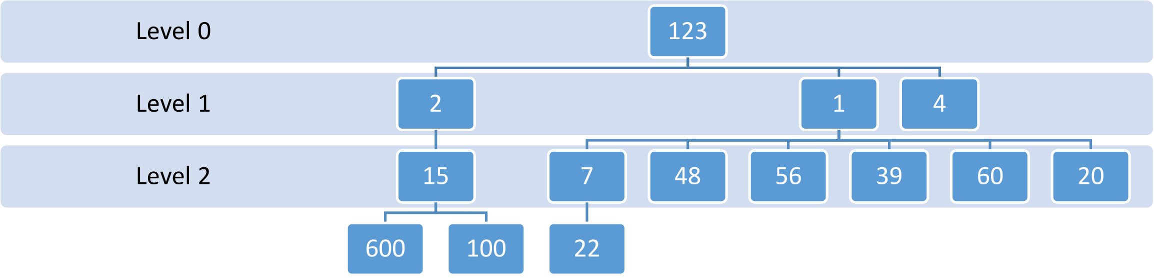
Problem Statement

Write a procedure that will construct a tree with arbitrary branching (i.e. number of children per node is not known before construction), given a list Ls of non-negative numbers with 0s treated as separators. The tree is to be constructed in a depth-first, left-to-right fashion: each 0 terminates a path. Note that a path is a sequence of nodes from the root to a leaf i.e. consecutive 0s indicate NULL nodes (or no more children for the parent).

For instance, the sequence 1 15 2 15 0 0 1 0 7 0 12 0 346 0 0 22 23 0 0 7 252 0 41 0 17 0 7 0 0 34 1 0 130 0 0 0 would produce the following tree:



and the sequence 123 2 15 600 0 100 0 0 0 1 7 22 0 0 48 0 56 0 39 0 60 0 20 0 0 4 0 0 would produce the following tree:



In this problem, you are supposed to construct **three** such trees. The sequence for all the trees shall be given as input. Also, you must implement a function to print the breadth first order (left to right) of a given tree, separated by spaces. You can refer to the input format and the sample test case given below.

Input format

Each line will start with one of the following key values (0, 1, -1). Each key corresponds to a function and has an input pattern. Implement following functions according to the given input pattern and write a driver function which can call the functions as per the given key value.

Key	Function to call	Format	Description
0	createTrees	0 <Sequence_1> <Sequence_2> <Sequence_3>	"0" shows creation of three n-ary trees. <Sequence_1>, <Sequence_2> and <Sequence_3> will be sets of non-negative integers separated by spaces. You must read these three input sequences and convert them into trees as explained before.
1	printBFS	1 X	"X" is the ID of the tree. For e.g. X=1 represents the first tree created in the above function. The value of X can be 1, 2 or 3. This function prints Xth tree in **Breadth First Order** (left to right), separated by space.
-1	stop		stop the program

Test Case 1

Input	0
	1 2 16 0 20 0 0 3 0 7 0 0
	1 3 0 2 20 0 16 0 0 7 0 0
	1 7 0 2 0 3 20 0 16 0 0 0
	1 1
	-1

Output	1 2 3 7 16 20